# MySQL

## Mysql引擎

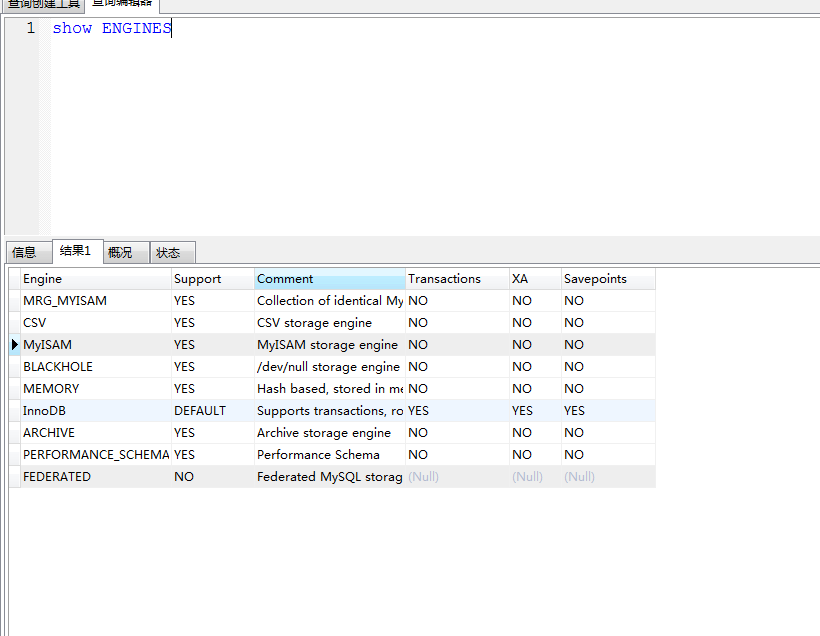
### 概述

   MySQL数 据库引擎取决于MySQL在安装的时候是如何被编译的。要添加一个新的引擎，就必须重新编译MYSQL。在缺省情况下，MYSQL支持三个引 擎：ISAM、MYISAM和HEAP。另外两种类型INNODB和BERKLEY（BDB），也常常可以使用。如果技术高超，还可以使用MySQL++ API自己做一个引擎。

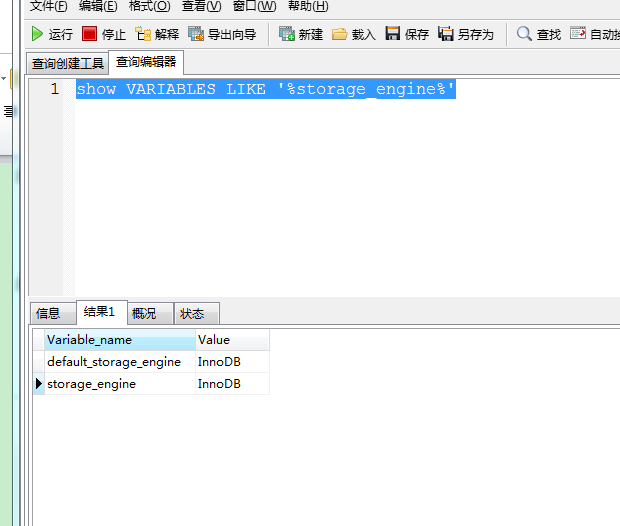
一般来说，MyISAM适合：(1)做很多count 的计算；(2)插入不频繁，查询非常频繁；(3)没有事务。InnoDB适合：(1)可靠性要求比较高，或者要求事务；(2)表更新和查询都相当的频繁，并且表锁定的机会比较大的情况。

一般情况下，MySQL会默认提供多种[存储引擎](http://wenwen.soso.com/z/Search.e?sp=S%E5%AD%98%E5%82%A8%E5%BC%95%E6%93%8E&ch=w.search.yjjlink&cid=w.search.yjjlink)，可以通过下面的查看:

1. 看你的MySQL现在已提供什么存储[引擎](http://wenwen.soso.com/z/Search.e?sp=S%E5%BC%95%E6%93%8E&ch=w.search.yjjlink&cid=w.search.yjjlink): mysql> show engines;



1. 看你的MySQL当前默认的存储引擎: mysql> show variables like '%storage\_engine%';



（1）加了索引以后，对于MyISAM查询可以加快：4 206.09733倍，对InnoDB查询加快510.72921倍，同时对MyISAM更新速度减慢为原来的1/2，InnoDB的更新速度减慢为原来的1/30。要看情况决定是否要加索引，比如不查询的log表，不要做任何的索引。

    （2）如果你的数据量是百万级别的，并且没有任何的事务处理，那么用MyISAM是性能最好的选择。

    （3）InnoDB表的大小更加的大，用MyISAM可省很多的硬盘空间。

### **ISAM**

ISAM是一个定义明确且历经时间考验的数据表格管理方法，它在设计之时就考虑到 数据库被查询的次数要远大于更新的次数。因此，ISAM执行读取操作的速度很快，而且不占用大量的内存和存储资源。ISAM的两个主要不足之处在于，它不 支持事务处理，也不能够容错：如果你的硬盘崩溃了，那么数据文件就无法恢复了。如果你正在把ISAM用在关键任务应用程序里，那就必须经常备份你所有的实 时数据，通过其复制特性，MYSQL能够支持这样的备份应用程序。

### MyISAM

MyISAM是MySQL的ISAM扩展格式和缺省的数据库引擎。除了提供ISAM里所没有的索引和字段管理的大量功能，MyISAM还使用一种表格锁定的机制，来优化多个并发的读写操作，其代价是你需要经常运行OPTIMIZE TABLE命令，来恢复被更新机制所浪费的空间。MyISAM还有一些有用的扩展，例如用来修复数据库文件的MyISAMCHK工具和用来恢复浪费空间的 MyISAMPACK工具。MYISAM强调了快速读取操作，这可能就是为什么MySQL受到了WEB开发如此青睐的主要原因：在WEB开发中你所进行的大量数据操作都是读取操作。所以，大多数虚拟主机提供商和INTERNET平台提供商只允许使用MYISAM格式。MyISAM格式的一个重要缺陷就是不能在表损坏后恢复数据。

### HEAP

HEAP允许只驻留在内存里的临时表格。驻留在内存里让HEAP要比ISAM和MYISAM都快，但是它所管理的数据是不稳定的，而且如果在关机之前没有进行保存，那么所有的数据都会丢失。在数据行被删除的时候，HEAP也不会浪费大量的空间。HEAP表格在你需要使用SELECT表达式来选择和操控数据的时候非常有用。要记住，在用完表格之后就删除表格。

### InnoDB

InnoDB数据库引擎都是造就MySQL灵活性的技术的直接产品，这项技术就是MYSQL++ API。在 使用MYSQL的时候，你所面对的每一个挑战几乎都源于ISAM和MyISAM数据库引擎不支持事务处理（transaction process）也不支持外来键。尽管要比ISAM和 MyISAM引擎慢很多，但是InnoDB包括了对事务处理和外来键的支持，这两点都是前两个引擎所没有的。如前所述，如果你的设计需要这些特性中的一者 或者两者，那你就要被迫使用后两个引擎中的一个了。

  MySQL 官方对InnoDB是这样解释的：InnoDB给MySQL提供了具有提交、回滚和崩溃恢复能力的事务安全（ACID兼容）存储引擎。 InnoDB锁定在行级并且也在SELECT语句提供一个Oracle风格一致的非锁定读，这些特色增加了多用户部署和性能。没有在InnoDB中扩大锁 定的需要，因为在InnoDB中行级锁定适合非常小的空间。InnoDB也支持FOREIGN KEY强制。在SQL查询中，你可以自由地将InnoDB类型的表与其它MySQL的表的类型混合起来，甚至在同一个查询中也可以混合。

    InnoDB是为处理巨大数据量时的最大性能设计，它的CPU效率可能是任何其它基于磁盘的关系数据库引擎所不能匹敌的。

    InnoDB存储引擎被完全与MySQL服务器整合，InnoDB存储引擎为在主内存中缓存数据和索引而维持它自己的缓冲池。InnoDB存储它的表＆索引在一个表空间中，表空间可以包含数个文件（或原始磁盘分区）。这与MyISAM表不同，比如在MyISAM表中每个表被存在分离的文件中。InnoDB 表可以是任何尺寸，即使在文件尺寸被限制为2GB的操作系统上。

    InnoDB默认地被包含在MySQL二进制分发中。Windows Essentials installer使InnoDB成为Windows上MySQL的默认表。

InnoDB被用来在众多需要高性能的大型数据库站点上产生。著名的Internet新闻站点Slashdot.org运行在InnoDB上。 Mytrix, Inc.在InnoDB上存储超过1TB的数据，还有一些其它站点在InnoDB上处理平均每秒800次插入/更新的

## 保留字

### 异常栈

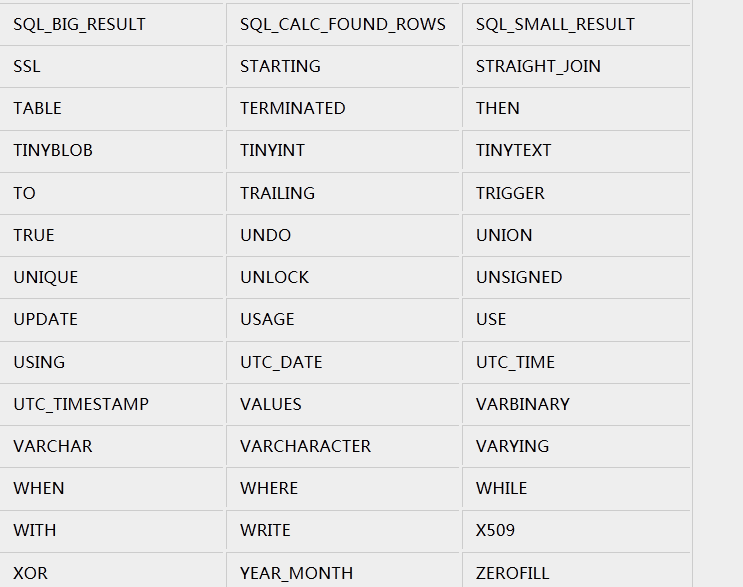
SQL Error: 1064, SQLState: 42000- You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near

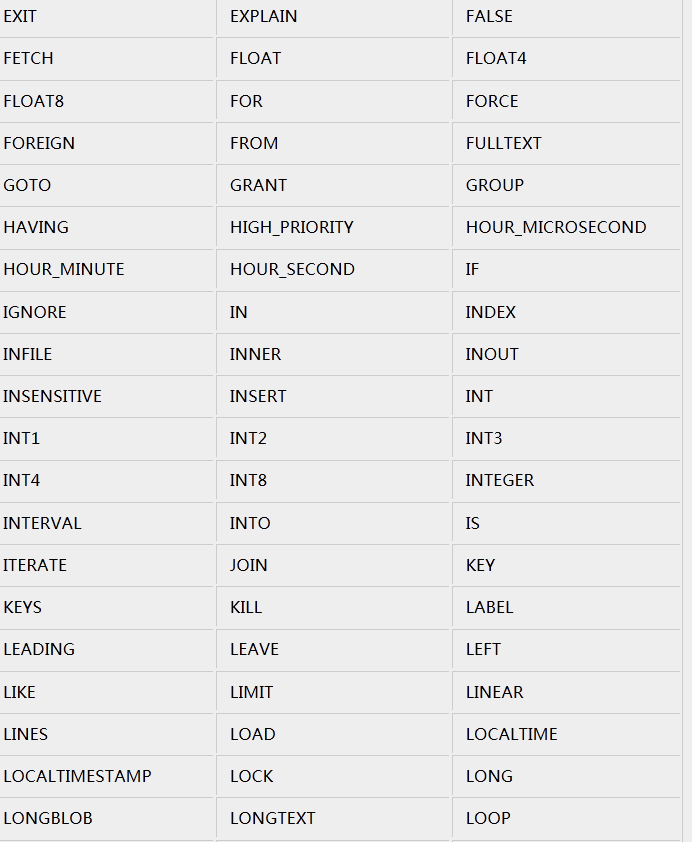
当出现类似于此的异常栈时时因为定义的字段和mysql的保留字有冲突

### Mysql保留字列表









### Mysql进程监控

#### 显示当前的进程列表

#### SHOW PROCESSLIST;

#### 查询当前某用户的进程id

拼接命令 复制出结果集执行后会杀死某用户的所有进程

SELECT CONCAT("KILL ",id ,";") FROM information\_schema.`PROCESSLIST` where `USER`='root'

## MySQL: InnoDB 还是 MyISAM?

MyISAM 是MySQL中默认的存储引擎，一般来说不是有太多人关心这个东西。决定使用什么样的存储引擎是一个很tricky的事情，但是还是值我们去研究一下，这里的文章只考虑 MyISAM 和InnoDB这两个，因为这两个是最常见的。

下面先让我们回答一些问题：

* 你的数据库有外键吗？
* 你需要事务支持吗？
* 你需要全文索引吗？
* 你经常使用什么样的查询模式？
*  你的数据有多大？

思考上面这些问题可以让你找到合适的方向，但那并不是绝对的。如果你需要事务处理或是外键，那么InnoDB 可能是比较好的方式。如果你需要全文索引，那么通常来说 MyISAM是好的选择，因为这是系统内建的，然而，我们其实并不会经常地去测试两百万行记录。所以，就算是慢一点，我们可以通过使用Sphinx从 InnoDB中获得全文索引。

数据的大小，是一个影响你选择什么样存储引擎的重要因素，大尺寸的数据集趋向于选择InnoDB方式，因为其支持事务处理和故障恢复。数据库的在小 决定了故障恢复的时间长短，InnoDB可以利用事务日志进行数据恢复，这会比较快。而MyISAM可能会需要几个小时甚至几天来干这些事，InnoDB 只需要几分钟。

您操作数据库表的习惯可能也会是一个对性能影响很大的因素。比如： COUNT() 在 MyISAM 表中会非常快，而在InnoDB 表下可能会很痛苦。而主键查询则在InnoDB下会相当相当的快，但需要小心的是如果我们的主键太长了也会导致性能问题。大批的inserts 语句在MyISAM下会快一些，但是updates 在InnoDB 下会更快一些——尤其在并发量大的时候。

所以，到底你检使用哪一个呢？根据经验来看，如果是一些小型的应用或项目，那么MyISAM 也许会更适合。当然，在大型的环境下使用MyISAM 也会有很大成功的时候，但却不总是这样的。如果你正在计划使用一个超大数据量的项目，而且需要事务处理或外键支持，那么你真的应该直接使用InnoDB方 式。但需要记住InnoDB 的表需要更多的内存和存储，转换100GB 的MyISAM 表到InnoDB 表可能会让你有非常坏的体验。

## Mysql优化

今天，数据库的操作越来越成为整个应用的性能瓶颈了，这点对于Web应用尤其明显。关于数据库的性能，这并不只是DBA才需要担心的事，而这更是我们程序 员需要去关注的事情。当我们去设计数据库表结构，对操作数据库时（尤其是查表时的SQL语句），我们都需要注意数据操作的性能。这里，我们不会讲过多的 SQL语句的优化，而只是针对MySQL这一Web应用最多的数据库